ANYLAYER: ORDER-AGNOSTIC LAYER DECOMPOSITION WITH HETEROGENEOUS QUERIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent approaches to image layer decomposition have made substantial progress. However, existing methods typically support limited query modalities or process targets independently, making it difficult to jointly handle heterogeneous queries while maintaining stable query–layer correspondence. We present AnyLayer, a unified framework that maps heterogeneous text, point, box, and mask queries to their corresponding RGBA foreground layers and a background image within a single diffusion process. First, AnyLayer provides a unified prompting paradigm for jointly processing heterogeneous queries. Second, it introduces an order-agnostic layer representation that removes ordinal foreground identities while preserving explicit query–layer correspondence. Third, Hybrid Layer Attention improves multi-layer generation efficiency by reducing redundant cross-layer computation. Experiments demonstrate effective mixed-modality control, robust query–layer association under increasing query cardinality and query permutations, competitive foreground and background quality, and efficient joint multi-layer inference.

1 INTRODUCTION

Modern image editing workflows operate on layers, yet most available images are flattened RGB observations. Recovering editable layers from such an image is fundamentally different from segmenting visible objects: the model must identify user-specified targets, infer their occluded appearance, estimate continuous transparency, and generate the background after removing the queried targets.

Given an input image $\mathbf{I}$ and a collection of distinct target queries $\mathcal{Q} = \{q_i\}_{i=1}^{N}$, our goal is to predict

$$\mathcal{F}(\mathbf{I}, \mathcal{Q}) = \left(\mathbf{I}_{\mathrm{bg}}, \{\mathbf{I}_{\mathrm{fg},i}\}_{i=1}^{N}\right), \tag{1}$$

where $\mathbf{I}_{\mathrm{fg},i}$ is the complete RGBA foreground layer specified by q_i, and $\mathbf{I}_{\mathrm{bg}}$ is the background obtained after removing all queried targets while retaining unqueried objects. We consider both exhaustive and partial query sets, with each query identifying a distinct foreground layer. The predicted foregrounds are aligned with the input queries.

The query modalities provide different levels of conditioning. Text and point queries require the model to identify the target and estimate its complete generation support, whereas box and mask queries directly provide a support-inclusive region covering the potential complete target extent. Accordingly, the box/mask setting evaluates layer generation within a provided spatial domain rather than hidden-shape inference from visible-only prompts.

Recent generative decomposition methods recover hidden content and transparent layers (Zheng et al., 2021; Yang et al., 2025; Wang et al., 2025; Chen et al., 2026; Wang et al., 2026), while variable-layer models relax fixed output cardinality (Yin et al., 2026; Tang et al., 2026). Referring Layer Decomposition (Chen et al., 2026) further allows a target layer to be specified through a flexible referring prompt. However, existing methods typically support limited query modalities, process targets independently, or lack an explicit mechanism for associating multiple heterogeneous queries with multiple generated outputs. We therefore study simultaneous multi-query, multi-output heterogeneous layer decomposition with explicit query–layer association.

A single-target decomposer, such as RLD (Chen et al., 2026), could be applied independently to each query, thereby avoiding multi-query association within a single generation process. However,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

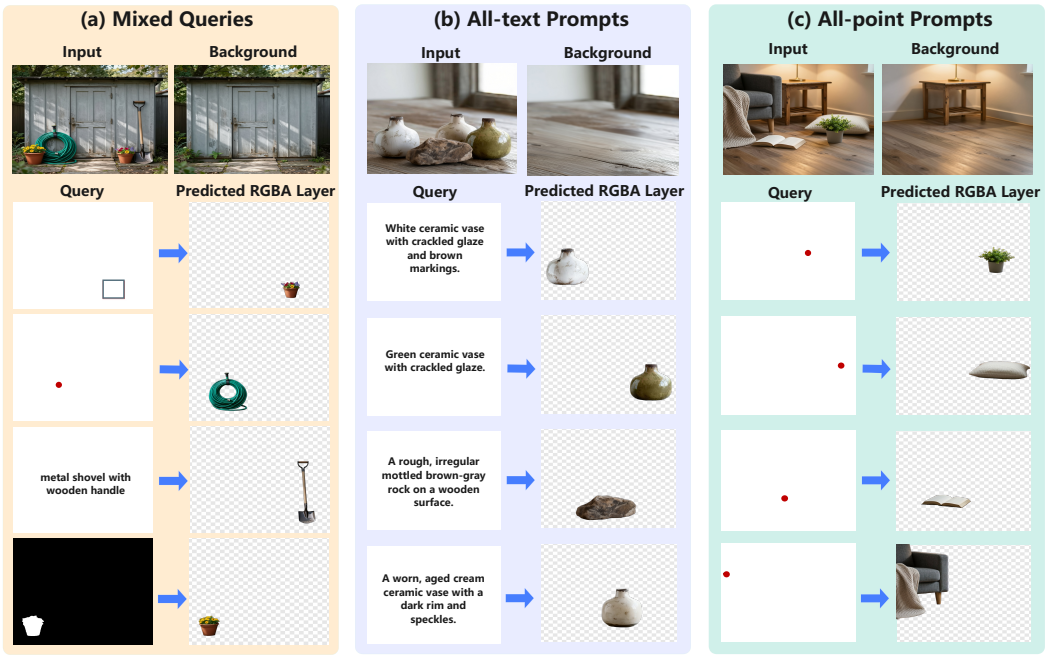


Figure 1: **Multi-query layer decomposition results of AnyLayer.** Given an input image, AnyLayer jointly predicts the queried RGBA foreground layers and a shared background using (a) a combination of box, point, text, and mask prompts, (b) text descriptions alone, or (c) point locations alone.

this strategy has two limitations. First, each run produces only one foreground layer and requires a separate denoising process, causing the total inference time to grow approximately linearly with the number of targets. Second, independently processing multiple targets does not directly produce a unified background with all queried targets removed; obtaining such a background requires additional reconciliation or sequential removal, which may introduce inconsistencies and accumulated errors. Joint generation instead produces multiple foreground layers and a shared background in a single diffusion process while amortizing computation across queries. These advantages motivate our parallel multi-output formulation, which introduces the following challenges.

First, heterogeneous queries convey target identity through semantics, locations, or spatial regions. Joint processing must preserve these distinct signals within a unified interface. While prior methods use dedicated point encoders (Kirillov et al., 2023; Ravi et al., 2025) or represent coordinates as language-like tokens (Yao et al., 2022), we serialize points as coordinate text and introduce Point-Grounded Text RoPE to explicitly anchor these tokens in the visual positional space.

Second, joint generation requires stable query–layer association. Index-dependent layer coordinates introduce ordinal dependence and empirically degrade performance at larger query counts. Simply sharing foreground coordinates, however, can cause target mixing without explicit query binding. We combine Layer-aligned Shared Position Encoding with Prompt-conditioned Attention Routing to remove ordinal identities while binding each foreground stream to its query and reference-image evidence.

Third, joint generation becomes costly as foreground tokens accumulate. We use a lightweight query-region predictor to select compact foreground supports for text and point queries, avoiding full-grid allocation. Hybrid Layer Attention further restricts direct cross-layer interactions to spatially aligned local neighborhoods in selected Transformer blocks, preserving cross-layer context while reducing attention computation.

To address these challenges, we present AnyLayer, a unified framework for simultaneous heterogeneous-query layer decomposition. As illustrated in Figure 1, given a flattened image and an arbitrarily ordered collection of queries, AnyLayer generates the corresponding complete RGBA foreground layers and a background image within a single diffusion trajectory. It supports both

homogeneous query sets and arbitrary combinations of text, point, box, and mask queries. By combining unified heterogeneous prompting, order-agnostic query–layer association, and Hybrid Layer Attention, AnyLayer enables target-specific multi-output generation while retaining the consistency and computational benefits of joint processing.

Our main contributions are:

- **Unified heterogeneous prompting.** We introduce a single conditional interface for jointly processing text, point, box, and mask queries, with Point-Grounded Text RoPE providing explicit spatial grounding for point queries.
- **Order-agnostic query–layer association.** Layer-aligned Shared Position Encoding removes ordinal foreground identities, while Prompt-conditioned Attention Routing preserves explicit query–layer correspondence under variable query counts and orderings.
- **Hybrid Layer Attention.** We reduce redundant cross-layer attention by processing foregrounds independently in most Transformer blocks while retaining spatially aligned local interactions at selected stages, improving multi-layer inference efficiency without substantially degrading decomposition quality.

2 Related Work

Layer decomposition and occlusion completion. Amodal segmentation estimates the complete spatial support of objects behind occlusion (Zhu et al., 2017; Qi et al., 2019; Zhan et al., 2020; Ke et al., 2021), but does not necessarily reconstruct their hidden appearance. Layer-by-layer scene decomposition extends this objective by recovering both visible and occluded content (Zheng et al., 2021). Recent generative methods leverage diffusion priors to recover complete layers. LayerDecomp addresses visual effects and alpha-composited content (Yang et al., 2025), while DiffDecompose performs diffusion-based layer-wise decomposition (Wang et al., 2025). Referring Layer Decomposition (RLD/RefLayer) maps a flexible referring prompt to a complete RGBA layer (Chen et al., 2026). RLD primarily processes one referred target at a time, whereas we jointly map multiple heterogeneous queries to their corresponding RGBA layers and a background image.

Variable-cardinality layered generation. Qwen-Image-Layered introduces variable-layer generation and decomposition (Yin et al., 2026); MRT uses masked region tokens for scalable layered generation and editing (Tang et al., 2026); LayerDiff generates composable transparent layers (Huang et al., 2024); and RevealLayer disentangles visible and hidden regions in multi-layer decomposition (Wang et al., 2026). These methods advance multi-layer generation and decomposition, whereas our focus is the foreground-order bias and explicit query–layer association that arise when multiple heterogeneous queries are jointly mapped to multiple outputs.

Prompt conditioning. Referring segmentation and promptable segmentation support language, point, box, or mask guidance for predicting visible regions (Hu et al., 2016; Wang et al., 2022; Yang et al., 2022; Kirillov et al., 2023; Ravi et al., 2025), but do not recover complete RGBA layers. Related vision–language methods represent spatial coordinates as discrete tokens (Yao et al., 2022); in contrast, we ground coordinate-bearing query text directly in the visual RoPE coordinate system (Su et al., 2024).

Matting and transparent generation. Learned matting methods estimate continuous opacity from observable foregrounds (Xu et al., 2017; Lu et al., 2019; Li & Lu, 2020; Yao et al., 2024b), while promptable extensions combine matting with segmentation foundations (Li et al., 2024; Yao et al., 2024a). Latent Transparency instead enables diffusion models to generate native transparent content (Zhang & Agrawala, 2024). In AnyLayer, an RGB-latent-compatible RGBA decoder provides continuous color and transparency outputs while preserving compatibility with the pretrained diffusion latent space.

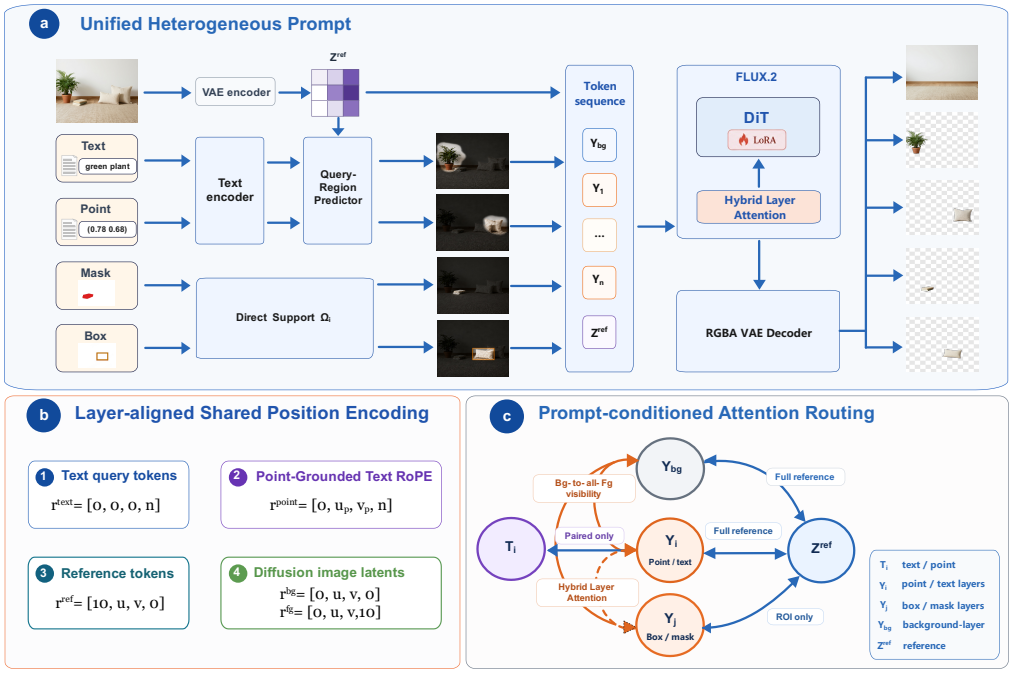


Figure 2: **Overview of AnyLayer.** (a) Heterogeneous queries define foreground supports for joint layer generation. (b) Position encoding provides point grounding and order-agnostic layer representations. (c) Attention routing establishes query–layer correspondence.

3 METHOD

3.1 OVERVIEW

Figure 2(a) illustrates the pipeline of AnyLayer. The reference image is first encoded by the VAE encoder into latents $\mathbf{Z}^{\text{ref}}$. Text descriptions and point coordinates are encoded as query-specific prompt blocks $\mathbf{T}_i$ and, together with the reference-image latents, passed to the query-region predictor to estimate Ω_i, whereas box and mask prompts define Ω_i directly. Here, Ω_i denotes the foreground generation support, including the region to be completed.

The prompt blocks and the selected foreground, background, and reference-image tokens are concatenated to form the joint input sequence at diffusion timestep t:

$$\mathbf{Z}_t = \left[[\mathbf{T}_i]_{i \in \mathcal{I}}; \mathbf{Y}_{\text{bg},t}; \mathbf{Y}_{1,t}; \ldots; \mathbf{Y}_{N,t}; \mathbf{Z}^{\text{ref}} \right]. \tag{2}$$

where $\mathcal{I}$ denotes the indices of text and point queries, $\mathbf{Y}_{\text{bg},t}$ is the complete noisy background grid, $\mathbf{Y}_{i,t}$ contains the noisy foreground tokens selected by Ω_i, and $\mathbf{Z}^{\text{ref}}$ denotes the complete reference-image latents.

The LoRA-adapted FLUX.2 Klein backbone jointly processes all tokens in $\mathbf{Z}_t$. Only the background and foreground tokens participate in the diffusion process. After denoising, the foreground tokens are scattered back to their original grid locations. The reconstructed foreground grids and the background grid are decoded by a separately trained RGBA decoder that remains frozen during diffusion training.

3.2 UNIFIED HETEROGENEOUS QUERY CONSTRUCTION

Each foreground query q_i is specified by a box, mask, text description, or point. A box is projected onto the latent grid, and its enclosed tokens form Ω_i. A mask is max-pooled onto the same grid, and its positive support forms Ω_i. A text query uses the block "Extract the foreground described

as <text>...</text> from the reference image." A point query is represented as the coordinate text <point>(x_i, y_i)</point>, where (x_i, y_i) denotes its normalized image coordinates. The resulting sequence is encoded as a query-specific text block.

As illustrated in Figure 2(b), we extend RoPE (Su et al., 2024) with four-axis indices to jointly encode token type, two-dimensional spatial location, and sequence position. An ordinary text token at within-block position n is assigned $[0, 0, 0, n]$. For the nth token of point query i, we use

$$\mathbf{r}_{i,n}^{\text{point}} = [0, u_i, v_i, n], \tag{3}$$

where (u_i, v_i) denotes the corresponding latent-grid location. This places the coordinate-bearing query tokens and visual tokens in the same two-dimensional positional coordinate system. We refer to this design as Point-Grounded Text RoPE.

3.2.1 Query-Region Prediction for Text and Point Queries

The predictor projects the reference-image latents $\mathbf{Z}^{\text{ref}}$ into a low-dimensional feature space and processes them with a one-layer image encoder. Each query-specific prompt block $\mathbf{T}_i$ is processed independently. A learned mask token is added to the beginning of $\mathbf{T}_i$, and two query-decoder layers read the encoded reference features. The decoded mask embedding is compared with the normalized visual embeddings to produce a region-probability map. The token-level support target derived from the complete foreground alpha is supervised using weighted binary cross-entropy and Dice losses:

$$\mathcal{L}_{\text{reg}} = \mathcal{L}_{\text{WBCE}} + \mathcal{L}_{\text{Dice}}. \tag{4}$$

A hard support Ω_i is obtained through thresholding and fixed post-processing.

3.3 Order-Agnostic Representation and Attention Routing

As illustrated in Figure 2(b), all foreground layers share the same layer-axis coordinate while retaining their original latent-grid positions. This Layer-aligned Shared Position Encoding removes modality-specific and ordinal foreground identities, avoiding dependence on the presentation order of the queries.

Shared coordinates alone cannot bind queries to their corresponding layers. We therefore introduce **Prompt-conditioned Attention Routing** (PAR), an attention-mask-based topology illustrated in Figure 2(c). Each foreground retains within-layer attention. Communication between different foregrounds is enabled only in selected Transformer blocks according to the Hybrid Layer Attention scheme described in Section 3.4. A text or point foreground attends to its paired query block, but not to other query-specific blocks, and accesses the full reference grid. The paired query block attends to itself and its corresponding foreground. A box or mask foreground accesses the reference tokens selected by its own spatial support. The background and reference tokens communicate with the foreground tokens.

3.4 Hybrid Layer Attention

Multi-layer decomposition benefits from cross-layer context for occlusion completion. However, excessive inter-layer interaction may introduce cross-target interference and substantially increase computation as the number of foregrounds grows. We therefore introduce **Hybrid Layer Attention** (HLA), which interleaves **Cross-Layer Attention** (CLA) and **Intra-Layer Attention** (ILA) to preserve necessary cross-layer communication while reducing redundant interactions.

As illustrated in Figure 3(a), CLA allows each foreground token to attend globally within its own layer while additionally accessing spatially aligned local neighborhoods in other foreground layers. In contrast, as shown in Figure 3(b), ILA blocks cross-layer communication and performs global attention independently within each foreground layer.

The two attention patterns follow a recurring 1:2 arrangement, with each Cross-Layer Neighborhood Attention block followed by two Intra-Layer Attention blocks. This hybrid topology retains periodic cross-layer communication while avoiding global cross-layer interactions in most Transformer blocks.

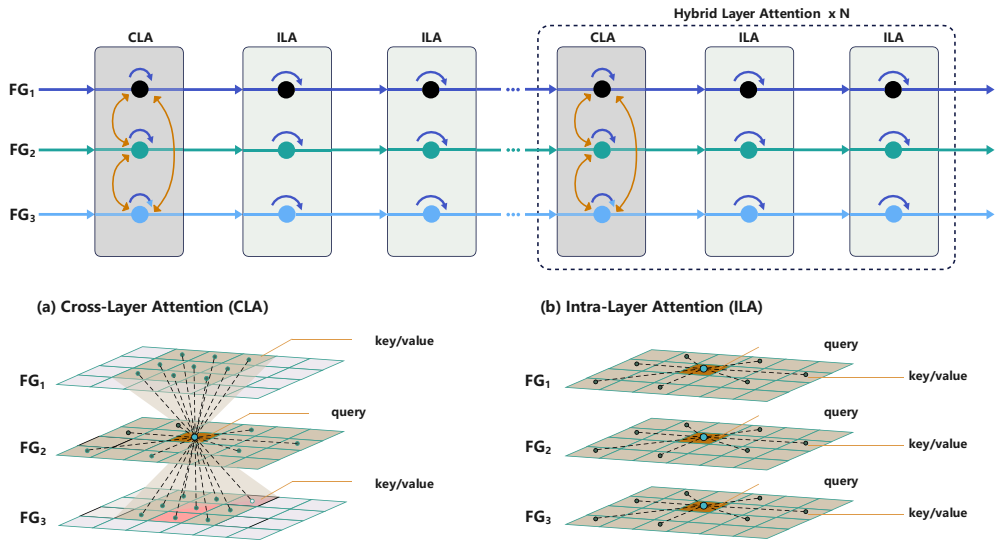


Figure 3: **Hybrid Layer Attention.** (a) Cross-layer attention accesses aligned local neighborhoods across foregrounds. (b) Intra-layer attention processes each foreground independently.

Diffusion training. We train AnyLayer using flow matching (Lipman et al., 2023). For each clean background or foreground latent $\mathbf{Y}_k$, where $k \in \{\text{bg}, 1, \ldots, N\}$, we sample $\boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $t \sim \mathcal{U}(0, 1)$, and construct

$$\mathbf{Y}_{k,t} = (1-t)\mathbf{Y}_k + t\boldsymbol{\epsilon}_k, \qquad \mathbf{u}_k = \boldsymbol{\epsilon}_k - \mathbf{Y}_k. \tag{5}$$

The per-layer flow loss $\mathcal{L}^k_{\text{flow}}$ is the token-normalized MSE between the predicted velocity and $\mathbf{u}_k$, preventing large supports from dominating training. The complete objective is

$$\mathcal{L} = \mathcal{L}^{\text{bg}}_{\text{flow}} + \frac{\lambda_{\text{fg}}}{N} \sum_{i=1}^{N} \mathcal{L}^i_{\text{flow}} + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}}, \tag{6}$$

where λ_{fg} and λ_{reg} weight the foreground and region-prediction losses, respectively.

RGBA decoding. Each ground-truth RGBA foreground is composited over a fixed gray background and encoded by the frozen RGB VAE. A separately trained decoder reconstructs straight RGB and continuous alpha using alpha-reconstruction and rendering-consistency losses. The decoder is frozen during diffusion training and decodes both foreground and background latents.

4 Experiments

4.1 Experimental Setup

Dataset. Training uses approximately 180K curated and synthetically composited layered samples with one to seven foreground layers and mixed text, point, box, and mask queries. We evaluate on RevealLayerBench and our 200-sample AnyLayerBench, which contains 5–15 foregrounds per image for high-cardinality evaluation. Dataset construction and prompt-generation details are provided in the supplementary material.

Implementation details. AnyLayer is built on FLUX.2 Klein with rank-128 LoRA adaptation and trained for 72 hours on 16 NVIDIA H800 GPUs. Complete optimization, preprocessing, and inference settings are provided in the supplementary material.

Table 1: **Comparison on RevealLayerBench.** Qwen-Image-Layered is conditioned only on the requested number of foreground layers; "–" denotes unavailable results.

		Foreground					Background			
Method	Query	PSNR↑	SSIM↑	LPIPS↓	FID↓	SoftIoU↑	PSNR↑	SSIM↑	LPIPS↓	FID↓
Qwen-Image-Layered	Native	–	–	–	–	–	16.87	0.7600	0.3437	136.1
RevealLayer	Box	32.36	0.9788	0.0214	17.82	0.9326	24.82	0.8410	0.1536	59.27
AnyLayer	Box	33.19	0.9811	0.0170	15.25	0.9390	25.51	0.8544	0.1359	44.99
AnyLayer	Text	30.63	0.9691	0.0437	20.00	0.8245	25.28	0.8547	0.1397	49.25
AnyLayer	Point	32.30	0.9792	0.0225	18.89	0.9052	25.04	0.8543	0.1419	51.08
AnyLayer	Mask	34.33	0.9830	0.0150	13.35	0.9496	25.58	0.8565	0.1359	44.45
AnyLayer	Mixed	33.00	0.9805	0.0196	16.25	0.9051	25.29	0.8544	0.1389	47.54



Figure 4: **Qualitative comparison under box queries.** Qwen-Image-Layered uses only the requested layer count and runs out of memory at high cardinality.

Evaluation protocol. We compare with RevealLayer (Wang et al., 2026) and Qwen-Image-Layered (Yin et al., 2026); the latter uses only the requested layer count. RLD/RefLayer (Chen et al., 2026) and MRT (Tang et al., 2026) are excluded due to unavailable public implementations. We evaluate foregrounds and backgrounds using PSNR, SSIM, LPIPS, and FID, and foreground alpha mattes using SoftIoU. RGBA foregrounds are rendered on a standardized background and scored over the full canvas. Efficiency is measured on the same NVIDIA H800 GPU.

4.2 Comparison with State of the Art

Table 1 shows that AnyLayer produces stable foreground and background outputs across all five query configurations. Box and mask queries obtain the strongest foreground results because they provide explicit regional evidence. Point queries remain competitive despite specifying only a single location, while text queries are more challenging because both target identity and spatial extent must be inferred from semantic descriptions. The mixed-modality setting performs close to the spatial-query settings, showing that text, point, box, and mask queries can be processed jointly within the same generation process. Background performance varies only moderately across query configurations. Qualitative comparisons under both low- and high-cardinality settings are shown in Figure 4.

4.3 Ablation Studies

We ablate the principal components of AnyLayer: Point-Grounded Text RoPE for point-query grounding, Layer-aligned Shared Position Encoding (LSPE) and Prompt-conditioned Attention Routing (PAR) for order-agnostic query–layer association, and the region predictor (RP) and Hybrid

Table 2: Ablation of PG-Text RoPE under the all-point query setting on RevealLayerBench.

Setting	FG PSNR↑	FG SSIM↑	FG LPIPS↓	FG FID↓	FG SoftIoU↑	BG PSNR↑	BG SSIM↑	BG LPIPS↓	BG FID↓
w/o PG-Text RoPE	29.41	0.9695	0.0423	27.44	0.7675	24.15	0.8426	0.1540	59.96
AnyLayer	**32.30**	**0.9792**	**0.0225**	**18.89**	**0.9052**	**25.04**	**0.8543**	**0.1419**	**51.08**

Table 3: Component ablation on the mixed-modality setting of RevealLayerBench.

	Components				Foreground					Background			
Variant	LSPE	PAR	RP	HLA	PSNR↑	SSIM↑	LPIPS↓	FID↓	SoftIoU↑	PSNR↑	SSIM↑	LPIPS↓	FID↓
w/o LSPE & PAR	–	–	✓	✓	30.18	0.9752	0.0255	22.50	0.9081	23.08	0.8292	0.1673	63.08
w/o PAR	✓	–	✓	✓	28.89	0.9740	0.0267	22.38	0.8830	22.32	0.7872	0.1739	69.00
w/o LSPE	–	✓	✓	✓	32.17	0.9786	0.0254	17.85	0.9007	24.09	0.8357	0.1492	49.51
w/o RP & HLA	✓	✓	–	–	32.77	0.9795	0.0210	16.84	0.9172	25.09	0.8543	0.1429	49.33
w/o HLA	✓	✓	✓	–	**33.08**	**0.9821**	**0.0190**	**16.20**	**0.9230**	25.17	0.8492	0.1423	48.52
Full model (sequential)	✓	✓	✓	✓	26.94	0.9753	0.0496	54.98	0.4945	12.64	0.5972	0.6811	175.50
Full model (joint)	✓	✓	✓	✓	33.00	0.9805	0.0196	16.25	0.9211	**25.29**	**0.8544**	**0.1389**	**47.54**

Table 4: Foreground quality under increasing query cardinality on the all-box setting of AnyLayer-Bench. All methods process 5, 10, or 15 foreground queries simultaneously.

	5 layers					10 layers					15 layers				
Method	PSNR↑	SSIM↑	LPIPS↓	FID↓	SoftIoU↑	PSNR↑	SSIM↑	LPIPS↓	FID↓	SoftIoU↑	PSNR↑	SSIM↑	LPIPS↓	FID↓	SoftIoU↑
RevealLayer	35.971	0.9947	0.0063	30.753	0.8280	35.993	0.9946	0.0061	22.433	0.8374	34.419	0.9911	0.0097	21.729	0.8100
w/o LSPE & PAR	35.879	0.9955	0.0058	28.995	0.8509	35.909	0.9955	0.0059	21.361	0.8448	34.505	0.9928	0.0085	19.753	0.8216
w/o PAR	32.112	0.9936	0.0135	67.326	0.5961	31.282	0.9928	0.0155	61.693	0.5869	28.413	0.9869	0.0252	78.955	0.4938
w/o LSPE	37.627	**0.9963**	0.0049	22.373	0.8756	36.560	0.9954	0.0050	17.828	0.8725	34.054	0.9932	0.0079	17.045	0.8507
AnyLayer	**37.758**	**0.9963**	**0.0048**	**21.810**	**0.8768**	**38.005**	**0.9964**	**0.0047**	**16.238**	**0.8765**	**36.554**	**0.9936**	**0.0072**	**15.476**	**0.8595**

Layer Attention (HLA) for efficient multi-layer inference. Unless otherwise specified, all variants use the same training data, optimization settings, and inference configuration.

Point-query grounding. We evaluate Point-Grounded Text RoPE (PG-Text RoPE) on the all-point evaluation set. The baseline represents point coordinates as query text with ordinary text positions, whereas the full model assigns the coordinate-bearing tokens to their corresponding locations in the visual RoPE coordinate system. As shown in Table 2, PG-Text RoPE consistently improves both foreground and background quality, demonstrating that explicit positional grounding enables point-query tokens to provide more effective spatial conditions.

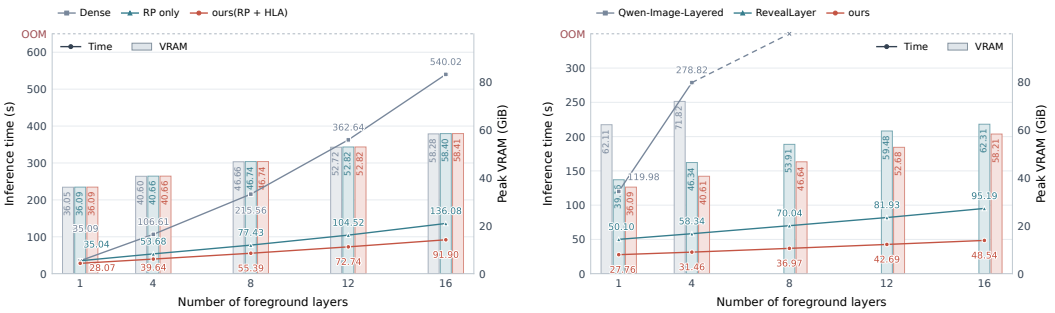
Order-agnostic query–layer association. Table 3 ablates LSPE and PAR on the mixed-modality setting of RevealLayerBench. Without LSPE, foregrounds use index-dependent layer coordinates; without PAR, explicit query-conditioned routing is removed. LSPE alone provides no improvement because shared foreground positions introduce ambiguity without explicit routing. PAR substantially improves target-specific generation, while combining both components achieves the best overall results. These results confirm their complementary roles: LSPE removes ordinal foreground identities, whereas PAR resolves the resulting positional symmetry and maintains query–layer association.

We further evaluate the model variants on AnyLayerBench with 5, 10, and 15 simultaneous box queries. As shown in Table 4, the full model consistently achieves the best foreground quality as query cardinality increases. Removing LSPE leads to a larger quality drop as query cardinality increases, while removing PAR substantially degrades foreground reconstruction across all evaluated cardinalities.

To evaluate sensitivity to query presentation order, we randomly permute each query together with its corresponding foreground support and initial noise tensor, while keeping the background and its initial noise unchanged. The generated foregrounds are then realigned by query identity before comparison. As shown in Table 5, the full model achieves the lowest foreground and background consistency MSE between the original-order and permuted-order predictions. Together, these results demonstrate robust query–layer association under increasing query cardinality and changes in query presentation order, independently of sampling-noise reassignment.

Table 5: **Robustness to query permutation on AnyLayerBench.** Queries, foreground supports, and corresponding noise tensors are permuted together, while the background and its noise remain fixed. Lower is better; "–" denotes unavailable results.

	All-box				Mixed-modality			
	Foreground MSE ($\times 10^{-4}$)		Background MSE ($\times 10^{-4}$)		Foreground MSE ($\times 10^{-4}$)		Background MSE ($\times 10^{-4}$)	
Method	Mean↓	Std.↓	Mean↓	Std.↓	Mean↓	Std.↓	Mean↓	Std.↓
RevealLayer	6.846	16.58	11.76	20.55	–	–	–	–
w/o LSPE & PAR	6.762	22.74	18.16	37.26	4.294	25.33	10.23	25.69
w/o PAR	16.52	23.95	16.35	43.51	11.42	26.91	9.202	18.61
w/o LSPE	4.424	27.27	5.051	12.45	3.787	15.34	7.371	23.96
AnyLayer	**1.694**	**4.557**	**1.853**	**3.980**	**0.9452**	**3.402**	**1.252**	**3.143**



(a) Efficiency ablation under mixed prompts.

(b) Method comparison on the all-box benchmark.

Figure 5: Inference efficiency with increasing foreground layer counts. Lines show inference time (left axis), and bars show peak allocated GPU memory (right axis).

Inference efficiency. Table 3 and Figure 5 show that RP and HLA improve efficiency while maintaining comparable quality. Sequential inference processes one query per pass and propagates the predicted background, causing substantial error accumulation and requiring repeated denoising. Joint inference instead produces all foregrounds and a shared background in one pass, substantially improving both quality and efficiency. RP reduces foreground tokens for text and point queries, while HLA reduces redundant cross-layer attention.

Figure 5(a) evaluates mixed-modality queries. At 16 layers, dense inference, RP, and RP+HLA require 540.02 s, 136.08 s, and 91.90 s, respectively, giving the full model a $5.88\times$ speedup over dense inference. Figure 5(b) evaluates all-box queries, where RP is not used. A single-query run requires 27.76 s, implying 444.16 s for 16 repeated runs, whereas joint 16-layer inference requires only 48.54 s. At the same cardinality, RevealLayer requires 95.19 s and 62.31 GiB, compared with 48.54 s and 58.21 GiB for AnyLayer; Qwen-Image-Layered runs out of memory at eight layers. These results demonstrate the scalability of joint inference and the complementary efficiency benefits of RP and HLA.

5 Conclusion

We presented AnyLayer, a unified framework that jointly maps heterogeneous text, point, box, and mask queries to multiple RGBA foreground layers and a background image. Unified prompting enables flexible target specification, while shared positional encoding and attention routing provide order-agnostic query–layer association. Hybrid Layer Attention further improves multi-layer inference efficiency. Experiments demonstrate effective heterogeneous-query control and stable performance across query orders and cardinalities. Performance may degrade under severe or complex occlusion. In addition, box and mask queries currently require the generation support to cover potential regions of completion. Future work will investigate more robust occlusion completion and automatic estimation of complete foreground support from visible prompts.

AI USE STATEMENT

Generative AI tools were used to assist with and polish the writing of this paper. In addition, GPT-Image was used for image generation and editing to construct the AnyLayerBench evaluation dataset.

REFERENCES

Fangyi Chen et al. Referring layer decomposition. In *ICLR*, 2026.

Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. In *ECCV*, 2016.

Runhui Huang et al. Layerdiff: Exploring text-guided multi-layered composable image synthesis via layer-collaborative diffusion model. *arXiv:2403.11929*, 2024.

Lei Ke, Yu-Wing Tai, and Chi-Keung Tang. Deep occlusion-aware instance segmentation with overlapping bilayers. In *CVPR*, 2021.

Alexander Kirillov et al. Segment anything. In *ICCV*, 2023.

Jiachen Li, Jitesh Jain, and Humphrey Shi. Matting anything. In *CVPR Workshops*, 2024.

Yaoyi Li and Hongtao Lu. Natural image matting via guided contextual attention. In *AAAI*, 2020.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Hao Lu et al. Indices matter: Learning to index for deep image matting. In *ICCV*, 2019.

Lu Qi et al. Amodal instance segmentation with kins dataset. In *CVPR*, 2019.

Nikhila Ravi et al. Sam 2: Segment anything in images and videos. In *ICLR*, 2025.

Jianlin Su et al. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Zhicong Tang et al. Mrt: Masked region transformer for layered image generation and editing at scale. In *CVPR*, 2026.

Binhao Wang et al. Reveallayer: Disentangling hidden and visible layers via occlusion-aware image decomposition. In *ICML*, 2026.

Zhaoqing Wang et al. Cris: Clip-driven referring image segmentation. In *CVPR*, 2022.

Zitong Wang et al. Diffdecompose: Layer-wise decomposition of alpha-composited images via diffusion transformers. *arXiv:2505.21541*, 2025.

Ning Xu et al. Deep image matting. In *CVPR*, 2017.

Jinrui Yang et al. Generative image layer decomposition with visual effects. In *CVPR*, 2025.

Zhao Yang et al. Lavt: Language-aware vision transformer for referring image segmentation. In *CVPR*, 2022.

Jingfeng Yao et al. Matte anything: Interactive natural image matting with segment anything models. *Image and Vision Computing*, 147:105067, 2024a.

Jingfeng Yao et al. Vitmatte: Boosting image matting with pretrained plain vision transformers. *Information Fusion*, 103:102091, 2024b.

Yuan Yao et al. Pevl: Position-enhanced pre-training and prompt tuning for vision-language models. In *EMNLP*, 2022.

Shengming Yin et al. Qwen-image-layered: Towards inherent editability via layer decomposition. In *CVPR*, 2026.

Xiaohang Zhan et al. Self-supervised scene de-occlusion. In *CVPR*, 2020.

Lvmin Zhang and Maneesh Agrawala. Transparent image layer diffusion using latent transparency. *ACM Transactions on Graphics*, 43(4), 2024.

Chuanxia Zheng et al. Visiting the invisible: Layer-by-layer completed scene decomposition. *IJCV*, 129:3195–3215, 2021.

Yan Zhu et al. Semantic amodal segmentation. In *CVPR*, 2017.